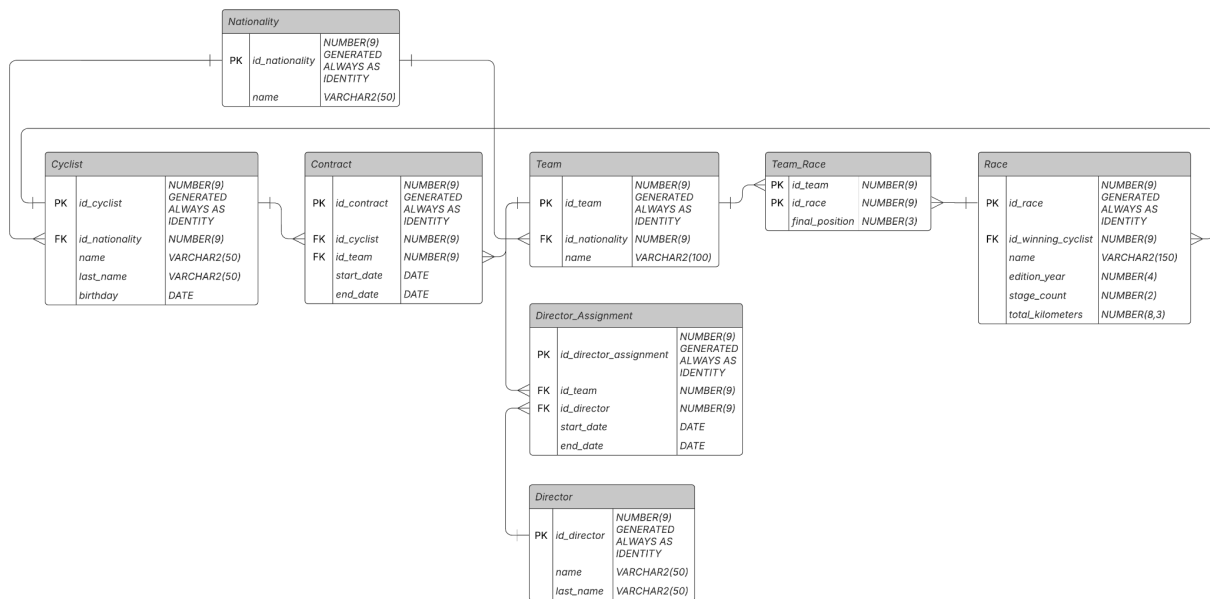


BASE DE DATOS II
TAREA DE LABORATORIO 01
TEMA: EJERCICIO DE TABLESPACE

Estudiante: Hanampa Bellido Iris Marisol

1. Elaborar el sizing de la base de datos a crear para un periodo de 5 años

1.1. Diagrama de la base de datos



1.2. Determinar el tamaño de la base de datos

Tipos de datos utilizados:

- NUMBER(9): Tamaño de 6 bytes
- NUMBER(8,3): Tamaño de 6 bytes
- NUMBER(4): Tamaño de 3 bytes
- NUMBER(3): Tamaño de 2 bytes
- NUMBER(2): Tamaño de 2 bytes
- VARCHAR2(50): Tamaño de 50 bytes
- VARCHAR2(100): Tamaño de 100 bytes
- DATE: Tamaño de 7 bytes

Tamaño de cada tabla:

Nationality

- id_nationality NUMBER(9) = 6 bytes
- name VARCHAR2(50) = 50 bytes

Total = 6 + 50 = 56 bytes

Director

- id_director NUMBER(9) = 6 bytes
- name VARCHAR2(50) = 50 bytes
- last_name VARCHAR2(50) = 50 bytes

Total = 6 + 50 + 50 = 106 bytes

Cyclist

- id_cyclist NUMBER(9) = 6 bytes
- id_nationality NUMBER(9) = 6 bytes
- name VARCHAR2(50) = 50 bytes
- last_name VARCHAR2(50) = 50 bytes
- birthday DATE = 7 bytes

Total = 6 + 6 + 50 + 50 + 7 = 119 bytes

Team

- id_team NUMBER(9) = 6 bytes
- id_nationality NUMBER(9) = 6 bytes
- name VARCHAR2(100) = 100 bytes

Total = 6 + 6 + 100 = 112 bytes

Race

- id_race NUMBER(9) = 6 bytes
- id_winning_cyclist NUMBER(9) = 6 bytes
- name VARCHAR2(150) = 150 bytes
- edition_year NUMBER(4) = 3 bytes
- stage_count NUMBER(2) = 2 bytes
- total_kilometers NUMBER(8,3) = 6 bytes

Total = 6 + 6 + 150 + 3 + 2 + 6 = 173 bytes

Team_Race

- id_team NUMBER(9) = 6 bytes
- id_race NUMBER(9) = 6 bytes
- final_position NUMBER(3) = 2 bytes

Total = 6 + 6 + 2 = 14 bytes

Director_Assignment

- id_director_assignment NUMBER(9) = 6 bytes
- id_team NUMBER(9) = 6 bytes
- id_director NUMBER(9) = 6 bytes

- start_date DATE = 7 bytes
- end_date DATE = 7 bytes

Total = 6 + 6 + 6 + 7 + 7 = 32 bytes

Contract

- id_contract NUMBER(9) = 6 bytes
- id_cyclist NUMBER(9) = 6 bytes
- id_team NUMBER(9) = 6 bytes
- start_date DATE = 7 bytes
- end_date DATE = 7 bytes

Total = 6 + 6 + 6 + 7 + 7 = 32 bytes

Tamaño total de todas las tablas:

56 + 106 + 119 + 112 + 173 + 14 + 32 + 32 = 644 bytes

1.3. Determinar una cantidad de registros iniciales por cada tabla

- Nationality: 150 registros iniciales
- Director: 100 registros iniciales
- Cyclist: 1000 registros iniciales
- Team: 100 registros iniciales
- Race: 200 registros iniciales
- Team_Race: 1000 registros iniciales
- Director_Assignment: 400 registros iniciales
- Contract: 2500 registros iniciales

1.4. Determinar la tasa de crecimiento y número de registros estimados a 5 años

Tabla	Año 1	Tasa anual (%)	Registros a 5 años
Nationality	150	5	$150 \times (5 + 10 \times 0.05) = 825$
Director	100	15	$100 \times (5 + 10 \times 0.15) = 650$
Cyclist	1,000	40	$1000 \times (5 + 10 \times 0.40) = 9,000$
Team	100	10	$100 \times (5 + 10 \times 0.10) = 600$
Race	200	20	$200 \times (5 + 10 \times 0.20) = 1,400$
Team_Race	1,000	30	$1000 \times (5 + 10 \times 0.30) = 8,000$
Director_Assignment	400	20	$400 \times (5 + 10 \times 0.20) = 2,800$
Contract	2,500	40	$2500 \times (5 + 10 \times 0.40) = 22,500$

Total			45,775
--------------	--	--	---------------

1.5. Determinar el tamaño en bytes para todas las tablas

Tabla	Tamaño fila (bytes)	Registros a 5 años	Tamaño total (bytes)
Nationality	56	825	46,200
Director	106	650	68,900
Cyclist	119	9,000	1,071,000
Team	112	600	67,200
Race	173	1,400	242,200
Team_Race	14	8,000	112,000
Director_Assignment	32	2,800	89,600
Contract	32	22,500	720,000
Total			2,416,100 bytes ≈ 2.42 MB

1.6. Determinar el tamaño de los índices

Nationality

- id_nationality: $6 \times 825 = 4,950$ bytes (Primary Key)

Total = 4,950 bytes

Director

- id_director: $6 \times 650 = 3,900$ bytes (Primary Key)
- name: $50 \times 650 = 32,500$ bytes (Non-unique, búsqueda por nombre)
- last_name: $50 \times 650 = 32,500$ bytes (Non-unique, búsqueda por apellido)

Total = 68,900 bytes

Cyclist

- id_cyclist: $6 \times 9,000 = 54,000$ bytes (Primary Key)
- id_nationality: $6 \times 9,000 = 54,000$ bytes (Foreign Key)
- name: $50 \times 9,000 = 450,000$ bytes (Non-unique, búsqueda por nombre)

- last_name: $50 \times 9,000 = 450,000$ bytes (Non-unique, búsqueda por apellido)

Total = 1,008,000 bytes

Team

- id_team: $6 \times 600 = 3,600$ bytes (Primary Key)
- id_nationality: $6 \times 600 = 3,600$ bytes (Foreign Key)
- name: $100 \times 600 = 60,000$ bytes (Non-unique, búsqueda por nombre de equipo)

Total = 67,200 bytes

Race

- id_race: $6 \times 1,400 = 8,400$ bytes (Primary Key)
- id_winning_cyclist: $6 \times 1,400 = 8,400$ bytes (Foreign Key)
- edition_year: $3 \times 1,400 = 4,200$ bytes (Non-unique, filtrar por año)
- id_winning_cyclist + edition_year: $(6+3) \times 1,400 = 12,600$ bytes (Non-unique, consultas por ganador y año)

Total = 33,600 bytes

Team_Race

- id_team + id_race: $(6+6) \times 8,000 = 96,000$ bytes (Primary Key)
- id_team: $6 \times 8,000 = 48,000$ bytes (Foreign Key)
- id_race: $6 \times 8,000 = 48,000$ bytes (Foreign Key)
- final_position: $2 \times 8,000 = 16,000$ bytes (Non-unique, consultas por posición final)

Total = 208,000 bytes

Director_Assignment

- id_director_assignment: $6 \times 995 = 5,970$ bytes (Primary Key)
- id_team: $6 \times 995 = 5,970$ bytes (Foreign Key)
- id_director: $6 \times 995 = 5,970$ bytes (Foreign Key)
- start_date: $7 \times 995 = 6,965$ bytes (Non-unique, filtrar asignaciones activas)
- end_date: $7 \times 995 = 6,965$ bytes (Non-unique, filtrar asignaciones activas)

Total = 31,840 bytes

Contract

- id_contract: $6 \times 22,500 = 135,000$ bytes (Primary Key)
- id_cyclist: $6 \times 22,500 = 135,000$ bytes (Foreign Key)

- `id_team`: $6 \times 22,500 = 135,000$ bytes (Foreign Key)
- `start_date`: $7 \times 22,500 = 157,500$ bytes (Non-unique, filtrar contratos vigentes)
- `end_date`: $7 \times 22,500 = 157,500$ bytes (Non-unique, filtrar contratos vigentes)
- `id_cyclist + id_team`: $(6+6) \times 22,500 = 270,000$ bytes (Non-unique, consultas por ciclista y equipo)

Total = 990,000 bytes

Tamaño total de todos los índices:

$56 + 106 + 119 + 112 + 173 + 14 + 32 + 32 = 2,412,590$ bytes (≈ 2.41 MB)

1.7. Determinar el espacio adicional para overhead

Consideraremos 25% de espacio para el overhead

1.8. Determinar el cálculo del sizing

- **Tamaño de tablas:** 2.42 MB
- **Tamaño de índices:** 2.41 MB
- **Total sin overhead:** $2.42 + 2.41 = 4.83$ MB
- **Con 25% de overhead:** $4.83 \times 1.25 = 6.04$ MB

Sizing estimado para 5 años: 6.04 MB.

2. Elaborar el tablespace para los datos y el tablespace temporal del usuario

2.1. Tablespace para datos

```
-- Tablespace para datos del sistema de ciclistas
CREATE TABLESPACE ts_data_cycling
DATAFILE 'ts_data_cycling.dbf' SIZE 6M
AUTOEXTEND ON NEXT 1M MAXSIZE 10M
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```

2.2. Tablespace temporal del usuario

```
-- Tablespace temporal del sistema de ciclistas
CREATE TEMPORARY TABLESPACE ts_temp_cycling
TEMPFILE 'ts_temp_cycling.dbf' SIZE 2M
AUTOEXTEND ON NEXT 1M MAXSIZE 5M
EXTENT MANAGEMENT LOCAL;
```

3. Elaborar el script en SQL de creación de los objetos de la base de datos (tablas, etc) en base al enunciado anterior.

- 3.1. Script para generar la base de datos

```
CREATE TABLE Nationality (  
  id_nationality NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
  name VARCHAR2(50),  
  PRIMARY KEY (id_nationality)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Director (  
  id_director NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
  name VARCHAR2(50),  
  last_name VARCHAR2(50),  
  PRIMARY KEY (id_director)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Cyclist (  
  id_cyclist NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
  id_nationality NUMBER(9),  
  name VARCHAR2(50),  
  last_name VARCHAR2(50),  
  birthday DATE,  
  PRIMARY KEY (id_cyclist),  
  CONSTRAINT FK_Cyclist_id_nationality  
    FOREIGN KEY (id_nationality)  
      REFERENCES Nationality(id_nationality)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Team (  
  id_team NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
  id_nationality NUMBER(9),  
  name VARCHAR2(100),  
  PRIMARY KEY (id_team),  
  CONSTRAINT FK_Team_id_nationality  
    FOREIGN KEY (id_nationality)  
      REFERENCES Nationality(id_nationality)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Race (  
  id_race NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
  id_winning_cyclist NUMBER(9),  
  name VARCHAR2(150),  
  edition_year NUMBER(4),  
  stage_count NUMBER(2),  
  total_kilometers NUMBER(8,3),  
  PRIMARY KEY (id_race),
```

```
CONSTRAINT FK_Race_id_winning_cyclist  
FOREIGN KEY (id_winning_cyclist)  
REFERENCES Cyclist(id_cyclist)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Team_Race (  
id_team NUMBER(9),  
id_race NUMBER(9),  
final_position NUMBER(3),  
PRIMARY KEY (id_team, id_race),  
CONSTRAINT FK_Team_Race_id_team  
FOREIGN KEY (id_team)  
REFERENCES Team(id_team),  
CONSTRAINT FK_Team_Race_id_race  
FOREIGN KEY (id_race)  
REFERENCES Race(id_race)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Director_Assignment (  
id_director_assignment NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
id_team NUMBER(9),  
id_director NUMBER(9),  
start_date DATE,  
end_date DATE,  
PRIMARY KEY (id_director_assignment),  
CONSTRAINT FK_Director_Assignment_id_team  
FOREIGN KEY (id_team)  
REFERENCES Team(id_team),  
CONSTRAINT FK_Director_Assignment_id_director  
FOREIGN KEY (id_director)  
REFERENCES Director(id_director)  
) TABLESPACE ts_data_cycling;
```

```
CREATE TABLE Contract (  
id_contract NUMBER(9) GENERATED ALWAYS AS IDENTITY,  
id_cyclist NUMBER(9),  
id_team NUMBER(9),  
start_date DATE,  
end_date DATE,  
PRIMARY KEY (id_contract),  
CONSTRAINT FK_Contract_id_team  
FOREIGN KEY (id_team)  
REFERENCES Team(id_team),  
CONSTRAINT FK_Contract_id_cyclist  
FOREIGN KEY (id_cyclist)  
REFERENCES Cyclist(id_cyclist)  
) TABLESPACE ts_data_cycling;
```

3.2. Script para crear los índices

```
-- Indices tabla Director
CREATE INDEX IDX_Director_name ON Director(name);
CREATE INDEX IDX_Director_last_name ON Director(last_name);

-- Indices tabla Cyclist
CREATE INDEX IDX_Cyclist_id_nationality ON Cyclist(id_nationality);
CREATE INDEX IDX_Cyclist_name ON Cyclist(name);
CREATE INDEX IDX_Cyclist_last_name ON Cyclist(last_name);

-- Indices tabla Team
CREATE INDEX IDX_Team_id_nationality ON Team(id_nationality);
CREATE INDEX IDX_Team_name ON Team(name);

-- Indices tabla Race
CREATE INDEX IDX_Race_id_winning_cyclist ON Race(id_winning_cyclist);
CREATE INDEX IDX_Race_edition_year ON Race(edition_year);
CREATE INDEX IDX_Race_winner_year ON Race(id_winning_cyclist,
edition_year);

-- Indices tabla Team_Race
CREATE INDEX IDX_Team_Race_id_team ON Team_Race(id_team);
CREATE INDEX IDX_Team_Race_id_race ON Team_Race(id_race);
CREATE INDEX IDX_Team_Race_final_position ON
Team_Race(final_position);

-- Indices tabla Director_Assignment
CREATE INDEX IDX_Director_Assignment_id_team ON
Director_Assignment(id_team);
CREATE INDEX IDX_Director_Assignment_id_director ON
Director_Assignment(id_director);
CREATE INDEX IDX_Director_Assignment_dates ON
Director_Assignment(start_date, end_date);

-- Indices tabla Contract
CREATE INDEX IDX_Contract_id_cyclist ON Contract(id_cyclist);
CREATE INDEX IDX_Contract_id_team ON Contract(id_team);
CREATE INDEX IDX_Contract_dates ON Contract(start_date, end_date);
CREATE INDEX IDX_Contract_cyclist_team ON Contract(id_cyclist,
id_team);
```

4. Elaborar el script en SQL de inserción y consultas para la base de datos.

4.1. Scripts de inserción

```
-- Inserciones en Nationality
```

```
INSERT INTO Nationality(name) VALUES ('Spain');
INSERT INTO Nationality(name) VALUES ('France');
INSERT INTO Nationality(name) VALUES ('Italy');
INSERT INTO Nationality(name) VALUES ('Colombia');
```

-- Inserciones en Director

```
INSERT INTO Director(name, last_name) VALUES ('John', 'Perez');
INSERT INTO Director(name, last_name) VALUES ('Michel', 'Durand');
INSERT INTO Director(name, last_name) VALUES ('Luca', 'Bianchi');
```

-- Inserciones en Cyclist

```
INSERT INTO Cyclist(id_nationality, name, last_name, birthday)
VALUES (1, 'Carlos', 'Martinez', DATE '1990-05-12');
INSERT INTO Cyclist(id_nationality, name, last_name, birthday)
VALUES (2, 'Pierre', 'Moreau', DATE '1988-09-23');
INSERT INTO Cyclist(id_nationality, name, last_name, birthday)
VALUES (4, 'Andres', 'Gomez', DATE '1995-02-14');
```

-- Inserciones en Team

```
INSERT INTO Team(id_nationality, name) VALUES (1, 'Team Sun');
INSERT INTO Team(id_nationality, name) VALUES (2, 'Team Light');
INSERT INTO Team(id_nationality, name) VALUES (4, 'Colombia Coffee');
```

-- Inserciones en Race

```
INSERT INTO Race(id_winning_cyclist, name, edition_year, stage_count,
total_kilometers)
VALUES (1, 'Vuelta a España', 2023, 21, 3300.500);
INSERT INTO Race(id_winning_cyclist, name, edition_year, stage_count,
total_kilometers)
VALUES (2, 'Tour de France', 2022, 21, 3400.200);
INSERT INTO Race(id_winning_cyclist, name, edition_year, stage_count,
total_kilometers)
VALUES (3, 'Giro d Italia', 2023, 21, 3450.700);
```

-- Inserciones en Team_Race

```
INSERT INTO Team_Race(id_team, id_race, final_position) VALUES (1, 1,
1);
INSERT INTO Team_Race(id_team, id_race, final_position) VALUES (2, 2,
2);
INSERT INTO Team_Race(id_team, id_race, final_position) VALUES (3, 3,
1);
```

-- Inserciones en Director_Assignment

```
INSERT INTO Director_Assignment(id_team, id_director, start_date,
end_date)
VALUES (1, 1, DATE '2020-01-01', DATE '2023-12-31');
INSERT INTO Director_Assignment(id_team, id_director, start_date,
end_date)
```

```
VALUES (2, 2, DATE '2021-01-01', NULL);
```

```
-- Inserciones en Contract
```

```
INSERT INTO Contract(id_cyclist, id_team, start_date, end_date)
```

```
VALUES (1, 1, DATE '2020-01-01', DATE '2022-12-31');
```

```
INSERT INTO Contract(id_cyclist, id_team, start_date, end_date)
```

```
VALUES (2, 2, DATE '2021-01-01', DATE '2023-12-31');
```

```
INSERT INTO Contract(id_cyclist, id_team, start_date, end_date)
```

```
VALUES (3, 3, DATE '2022-01-01', NULL);
```

4.2. Scripts de consultas

```
-- Consultar todos los ciclistas y sus nacionalidades
```

```
SELECT c.name AS first_name, c.last_name AS last_name, n.name AS  
nationality
```

```
FROM Cyclist c
```

```
JOIN Nationality n ON c.id_nationality = n.id_nationality;
```

```
-- Consultar los equipos y sus directores
```

```
SELECT t.name AS team, d.name AS director_first_name, d.last_name AS  
director_last_name
```

```
FROM Team t
```

```
JOIN Director_Assignment da ON t.id_team = da.id_team
```

```
JOIN Director d ON da.id_director = d.id_director;
```

```
-- Consultar las carreras y el ciclista ganador
```

```
SELECT r.name AS race, r.edition_year, c.name AS winner_first_name,  
c.last_name AS winner_last_name
```

```
FROM Race r
```

```
JOIN Cyclist c ON r.id_winning_cyclist = c.id_cyclist;
```

```
-- Consultar los contratos vigentes (end_date NULL o mayor a SYSDATE)
```

```
SELECT c.name AS cyclist_first_name, c.last_name AS cyclist_last_name,  
t.name AS team, ct.start_date, ct.end_date
```

```
FROM Contract ct
```

```
JOIN Cyclist c ON ct.id_cyclist = c.id_cyclist
```

```
JOIN Team t ON ct.id_team = t.id_team
```

```
WHERE ct.end_date IS NULL OR ct.end_date > SYSDATE;
```

```
-- Consultar posiciones finales de equipos en cada carrera
```

```
SELECT r.name AS race, r.edition_year, t.name AS team, tr.final_position
```

```
FROM Team_Race tr
```

```
JOIN Team t ON tr.id_team = t.id_team
```

```
JOIN Race r ON tr.id_race = r.id_race
```

```
ORDER BY r.edition_year DESC, tr.final_position ASC;
```